

1 Program Documentation

We implemented this assignment in Python, and our architecture is built off Simpy, a process-based discrete-event simulation library. Each source and the router is modelled as a processes while the transportation of packets are modelled as events. In our architecture we utilize a process interaction concept in Simpy known as Stores which allows the modelling of production and consumption of objects.

1.1 Routers

We took a class-based approach to implementing our router. We created a router superclass (in 'store_super.py'), FIFO and RR router classes subclassing it (in 'store_fifo.py' and 'store_rr.py' respectively), and a DRR router class subclassing the RR router class (in 'store_drr.py').

1.2 Sources

Simpy provides another process interaction called Containers which allow the modelling of quantities of similar discrete objects. In order to limit the number of packets used in an experiment we use a container and fill it with the number of packets we want to run. Each source will then decrement the value in this container each time it creates a packet. When The container is empty, sources will no longer be able to create packets and once the router transmits the last packet the simulation will terminate.

$$\sum \tag{1}$$

1.3 Running Instructions

1.3.1 Installation

In order to use Simpy on the csa2.bu.edu server we must use virtualenv to be able to install Python modules without root privilege in a sandbox environment. We created an install script that will download, install and configure our test environment so the experiments can be run.

At the command line run the following

```
./install.sh
```

If you are confronted with permission denied error you may need to change the permission on the file, at the terminal run:

```
chmod 755 install.sh
```

2 Experimental Results

2.1 FIFO Router

2.2 RR Router

2.3 DRR Router